

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСІЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ БІЗНЕСУ ТА СУЧАСНИХ
ТЕХНОЛОГІЙ**

**ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ
ІНФОРМАТИКИ**

Допускається до захисту

Завідувач кафедри _____ О.О. Ємець
(підпис)

«_____» _____ 2021 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

ДО БАКАЛАВРСЬКОЇ РОБОТИ

на тему

**СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТРЕНАЖЕРУ З ТЕМИ
«ЦИКЛ FOR В EXCEL VBA»
ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО КУРСУ
«ОРГАНІЗАЦІЯ ТА ОБРОБКА ЕЛЕКТРОННОЇ ІНФОРМАЦІЇ»**

зі спеціальності 122 «Комп'ютерні науки»

Виконавець роботи Малишева Дар'я Василівна

_____ «___» _____ 2021р.
(підпис)

Науковий керівник доц., каф.-мат. наук, Чілікіна Т.В.

_____ «___» _____ 2021р.
(підпис)

ПОЛТАВА 2021р.

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСІЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

ЗАТВЕРДЖУЮ

Завідувач кафедри _____ **О.О. Ємець**

« 8 » вересня 2020р.

**Завдання та календарний графік
виконання дипломної роботи**

**Студент(ка) спеціальності 122 «Комп'ютерні науки»
Прізвище, ім'я, по батькові Малишева Дар'я Василівна**

1. Тема «Створення програмного забезпечення тренажеру з теми «Цикл for в Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації» затверджена наказом ректора № 121-Н від « 1 » вересня 2020 р.

Термін подання студентом бакалаврської роботи « 20 » травня 2021 р.

2. Вихідні дані до дипломної роботи: публікації з теми навчальні тренажери в дистанційних курсах з комп'ютерних наук.

3. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

ВСТУП

1 Постановка задачі

2 Інформаційний огляд

2.1. Огляд схожих робіт

2.2. Позитивні аспекти розглянутих робіт.

2.3. Вади розробок розглянутих робіт.

2.4. Актуальність та необхідність розробки тренажера.

3 Теоретична частина

3.1 Теоретичні матеріали

3.1.1 Цикл For Next

3.1.2 Цикл For Loop

3.1.3 Цикл For Each...Next

3.1.4 Цикл «For Each»

3.1.5 Оператор переривання циклу «Exit For»

3.2 Алгоритм роботи тренажера

3.3 Блок-схема програми-тренажера

4 Практична частина

4.1 Опис програмного забезпечення

4.2 Перевірка валідності

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК А

4. Перелік графічного матеріалу: 1 аркуш блок-схем, інші необхідні ілюстрації.

5. Консультанти розділів бакалаврської роботи

Розділ	Прізвище, ініціали, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1. Постанова задачі	Чілікіна Т.В.	8.09.20	8.09.20
2. Інформаційний огляд	Чілікіна Т.В.	8.09.20	8.09.20
3. Теоретична частина	Чілікіна Т.В.	8.09.20	8.09.20
4. Практична реалізація	Чілікіна Т.В.	8.09.20	8.09.20

6. Календарний графік виконання бакалаврської роботи

Зміст роботи	Термін виконання	Фактичне виконання
1. Вступ	10.05.21	
2. Вивчення методичних рекомендацій та стандартів та звіт керівнику	15.09.20	
3. Постановка задачі	1.10.20	
4. Інформаційний огляд джерел бібліотек та інтернету	2.11.20	
5. Теоретична частина	1.02.21	
6. Практична частина	17.05.21	
7. Закінчення оформлення	21.05.21	
8. Доповідь студента на кафедрі	28.05.21	
9. Доробка (за необхідністю), рецензування	14.06.21	

Дата видачі завдання « 8 » вересня 2020 р.

Студент Малишева Дар'я Василівна

Науковий керівник _____ доц., каф.-мат. наук, Чілікіна Т.В.
(підпис)

Результати захисту бакалаврської роботи

Дипломна робота оцінена на _____
(балів, оцінка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № _____ від « _____ » _____ 2021 р.

Секретар ЕК _____
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Записка: 46 стор., в т.ч. основна частина 41 стор., джерел - 12.

Предмет розробки – програмне забезпечення з теми «Цикл for в Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації».

Мета роботи – програмно реалізувати програмне забезпечення з теми «Цикл for в Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації».

Методи роботи – програмна реалізація була проведена за допомогою платформи Unity Engine. Для створення програми було використано мову програмування C# та Microsoft Visual Studio.

КЛЮЧОВІ СЛОВА: ТРЕНАЖЕР, ЦИКЛИ, EXCEL VBA

ЗМІСТ

ВСТУП.....	4
1 Постановка задачі.....	7
2 Інформаційний огляд	8
2.1. Огляд схожих робіт	8
2.2. Позитивні аспекти розглянутих робіт.....	8
2.3. Вади розробок розглянутих робіт	8
2.4. Актуальність та необхідність розробки тренажера	8
3 Теоретична частина.....	9
3.1 Теоретичні матеріали	9
3.1.1 Цикл For Next	9
3.1.2 Цикл For Loop.....	11
3.1.3 Цикл For Each...Next.....	11
3.1.4 Цикл «For Each»	12
3.1.5 Оператор переривання циклу «Exit For».....	13
3.2 Алгоритм роботи тренажера.....	14
3.3 Блок-схема програми-тренажера.....	24
4 Практична частина	25
4.1 Опис програмного забезпечення	25
4.2 Перевірка валідності	29
ВИСНОВКИ	41
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	42
ДОДАТОК А	44

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, символів, скорочень
<i>ДК</i>	Дистанційний курс.
<i>ДН</i>	Дистанційне навчання.
<i>ООЕІ</i>	Організація та обробка електронної інформації
<i>Excel VBA</i>	Вбудована в Excel мова програмування

ВСТУП

В час бурхливого розвитку комп'ютерних технологій дистанційне навчання займає вадливу роль. Цей вид навчання надає можливість отримувати якісні знання за допомогою сучасних інформаційних та телекомунікаційних технологій, що дозволяють навчатись на відстані без безпосереднього, особистого контакту між викладачем і студентом.

Електронні комп'ютерні тренажери призначені для навчання, а також закріплення умінь та навичок розв'язування конкретних задач в відповідній предметній області. Тому є актуальним створення такого засобу дистанційного навчання.

Мета роботи – програмно реалізувати програмне забезпечення з теми «Цикл for в Excel VBA».

Об'єкт роботи – реалізація програмного забезпечення у вигляді тренажеру.

Предмет роботи – тренажер з теми «Цикл for в Excel VBA».

Методи роботи – програмна реалізація була проведена за допомогою платформи Unity Engine. Для створення програми було використано мову програмування C# та Microsoft Visual Studio.

Структура пояснювальної записки до бакалаврської роботи:

- титульний аркуш;
- завдання;
- реферат;
- зміст;
- вступ;
- теоретична частина;
- практична частина;
- висновки;
- рекомендації;
- список використаних джерел;

Обсяг пояснювальної записки: 46 стор., в т.ч. основна частина 41 стор., джерел - 12.

1 Постановка задачі

Основними завданнями даної бакалаврської роботи є:

1. Програмна реалізація елементів тренажеру з теми «Цикл for в Excel VBA».
2. Розробка алгоритму роботи з програмним забезпеченням.
3. Розробка блок-схем алгоритму роботи.

Основними вимогами до програмного забезпечення є:

1. Надійність програмного забезпечення.
2. Зручна навігація в програмному забезпеченні.
3. Наявність практичних завдань з теорією на одному екрані.

2 Інформаційний огляд

2.1. Огляд схожих робіт

Під час пошуку схожих робіт було обрано два тренажери, а саме тренажер на тему «Визначення рангу елемента системи» автор Тітар Є.В. [1], та тренажер з теми «Розв’язування дифрівнянь методом Ейлера» автор Цикозін О.В. [2].

2.2. Позитивні аспекти розглянутих робіт

Після проведення інформаційного огляду було отримано наступні позитивні аспекти з двох розглянутих робіт:

- Вибір складності прикладів;
- Повтор роботи без перезавантаження застосунку;
- Наявність тестів по теоретичному матеріалу;
- Перевірка введеної відповіді.

2.3. Вади розробок розглянутих робіт

Після проведення інформаційного огляду було отримано наступні вади двох розглянутих робіт:

- Відсутність тестових завдань по теоретичній частині;
- Наявність теоретичної інформації через вбудований файл .pdf;
- Відсутність підказок під час роботи.

2.4. Актуальність та необхідність розробки тренажера

Після аналізу обраних тренажерів було прийнято рішення покращити попередні розробки шляхом створення тренажера з теми «Цикл for в Excel VBA».

Створення тренажера, що виправить знайдені вади розглянутих тренажерів вважається актуальним та необхідним.

3 Теоретична частина

3.1 Теоретичні матеріали

Цикл For Loop в VBA - один з найпопулярніших циклів в Excel. Даний цикл має дві форми - For Next і For Each In Next. Дані оператори використовуються для послідовного переміщення по списку елементів або чисел. Для завершення циклу ми можемо в будь-який момент використовувати команду виходу.

3.1.1 Цикл For Next

Цикл For ... Next використовує змінну, яка послідовно приймає значення із заданого діапазону. З кожною зміною значення змінної виконуються дії, ув'язнені в тілі циклу. Це легко зрозуміти з простого прикладу:

```
For i = 1 To 10  
    Total = Total + iArray (i)  
Next i
```

У цьому простому циклі For ... Next використовується змінна i, яка послідовно приймає значення 1, 2, 3, ... 10, і для кожного з цих значень виконується код VBA, що знаходиться всередині циклу. Таким чином, даний цикл підсумовує елементи масиву iArray в змінної Total.

У наведеному вище прикладі крок збільшення циклу не вказано, тому для покрокового збільшення змінної i від 1 до 10 за замовчуванням використовується приріст 1. Однак, в деяких випадках потрібно використовувати інші значення збільшення для циклу. Це можна зробити за допомогою ключового слова Step, як показано в наступному простому прикладі.

```
For d = 0 To 10 Step 0.1  
    dTotal = dTotal + d  
Next d
```

Так як в наведеному вище прикладі заданий крок збільшення рівний 0.1, то змінна dTotal для кожного повторення циклу приймає значення 0.0, 0.1, 0.2, 0.3, ... 9.9, 10.0.

Для визначення кроку циклу в VBA можна використовувати негативну величину, наприклад, ось так:

```
For i = 10 To 1 Step -1
```

```
    iArray (i) = i
```

```
Next i
```

Тут крок збільшення дорівнює -1, тому змінна i з кожним повторенням циклу приймає значення 10, 9, 8, ... 1.

Даний цикл має наступний синтаксис:

```
For лічильник = початок_лічильника To кінець_лічильника
```

```
    'дія
```

```
Next лічильник
```

Те що ми робимо тут, по суті, це створюємо цикл, який використовує змінну лічильник як хранитель часу. Встановлюємо його значення рівним початок_лічильника, і збільшуємо (або зменшуємо) на 1 під час кожного витка. Цикл буде виконуватися до тих пір, поки значення лічильник не стане рівним кінець_лічильника. Коли обидва ці значення зійдуться, цикл виконається останній раз і зупиниться.

Приклад циклу:

```
Sub приклад_циклу1()
```

```
For лічильник = 1 to 10
```

```
    j = лічильник
```

```
Next лічильник
```

```
    msgbox "значення лічильника на останньому витку дорівнює " & j
```

```
End Sub
```

При останньому значенні змінної лічильник буде рівний 11.

3.1.2 Цикл For Loop

Якщо у вас з'явилася необхідність переміщатися від більшого значення до меншого - ви можете використовувати цикл в зворотному напрямку. Ось приклад зворотного циклу:

```
Sub приклад_циклу2()
```

```
For лічильник = 10 to 1 Step -1
```

```
  j = лічильник
```

```
Next лічильник
```

```
  msgbox " значення лічильника на останньому витку дорівнює " & j;  
  лічильник
```

```
End Sub
```

При останньому значенні змінної лічильник буде рівний 1.

Як ви могли помітити, ми можемо використовувати інструкцію Step n для роботи циклу як вперед, так і в зворотному напрямку. За замовчуванням значення Step дорівнює 1, але воно може бути змінено, якщо необхідно пропускати будь-які значення, тоді значення n буде більше одного, або переміщатися в зворотному напрямку, тоді n буде негативним.

3.1.3 Цикл For Each...Next

Даний цикл має наступний синтаксис:

```
For Each елемент_групи In група_елементів
```

```
  'Дія
```

```
Next елемент_групи
```

Тут змінна елемент_групи належить до групи_елементів. Мається на увазі, що об'єкт група_елементів повинен бути колекцією об'єктів. Ви не зможете запустити цикл For Each для окремого об'єкта (Microsoft відразу сповістить вас про це 438-ю помилкою).

Даний цикл перебирає всі елементи будь-якої колекції, починаючи з самого першого. Ви можете використовувати даний цикл, якщо вам необхідно, наприклад, обійти всі листи в книзі, об'єкти на аркуші, зведені таблиці і т.д.

Нижче представлений приклад, як можна скористатися циклом For Each для перегляду всіх листів книги:

```
Sub приклад_циклу4()
For Each sht In ActiveWorkbook.Worksheets
MsgBox sht.Name
Next sht
End Sub
```

Переривання циклу VBA

Якщо вам необхідно вийти з циклу до моменту, як буде досягнута умова завершення циклу, скористайтесь командою End For в зв'язці з інструкцією IF. У прикладі, наведеному нижче, ми вийдемо з циклу до моменту досягнення умови завершення, в даному циклі вихід буде здійснений за умови, коли змінна лічильник буде дорівнювати 3.

```
Sub приклад_циклу5()
For лічильник = 0 To 5
MsgBox лічильник
If (лічильник = 3) Then Exit For
Next лічильник
End Sub
```

Оператор циклу «For» в Visual Basic

Структура оператора циклу For в Visual Basic може бути організована в одній з двох форм: як цикл For ... Next або як цикл For Each.

3.1.4 Цикл «For Each»

Цикл For Each схожий на цикл For ... Next, але замість того, щоб перебирати послідовність значень для змінної-лічильника, цикл For Each виконує набір дій для кожного об'єкта із зазначеної групи об'єктів. У наступному прикладі за допомогою циклу For Each виконується перерахування всіх листів в поточній робочій книзі Excel:

```

Dim wSheet As Worksheet
For Each wSheet in Worksheets
    MsgBox "Знайдений лист:" & wSheet.Name
Next wSheet

```

3.1.5 Оператор переривання циклу «Exit For»

Оператор Exit For застосовується для переривання циклу. Як тільки в коді зустрічається цей оператор, програма завершує виконання циклу і переходить до виконання операторів, що знаходяться в коді відразу після цього циклу. Це можна використовувати, наприклад, для пошуку певного значення в масиві. Для цього за допомогою циклу проглядається кожен елемент масиву. Як тільки шуканий елемент знайдений, переглядати інші немає необхідності - цикл переривається.

Застосування оператора Exit For продемонстровано в наступному прикладі. Тут цикл перебирає 100 записів масиву і порівнює кожну зі значенням змінної dVal. Якщо збіг знайдено, то цикл переривається:

```

For i = 1 To 100
    If dValues (i) = dVal Then
        IndexVal = i
        Exit For
    End If
Next i

```


3.2 Алгоритм роботи тренажера

Під час роботи з тренажером після роботи з завданнями теоретичного вигляду користувач переходить до практичних завдань. Практичні завдання реалізовано через інпути, тобто поля для вводу інформації та через тести. Після вводу відповіді та натиснення на кнопку вводу користувач бачить свою введену відповідь та правильну відповідь. У повідомленні з помилкою, користувач отримує підказку з правильним варіантом відповіді.

Крок 1: Після запуску тренажера користувач переходить до головного меню з інформацією про тему роботи.

Крок 2: Після натиснення на кнопку «Перейти до роботи» користувач отримує теоретичний матеріал з теми.

Крок 3: Отримання теоретичного матеріалу:

«Для запобігання повторення роботи з однотипними завданнями можна використовувати вкладені в макроси Екселя цикли. За їх допомогою завдання, що зазвичай займають декілька годин можна зробити за кілька секунд.

В даному тренажері ми розглянемо два типи циклу For Next, а саме For Next Loops.»

Крок 4: Отримання теоретичного матеріалу:

«Цикл For Next дозволяє нам переглядати колекцію, елементами якої можуть бути об'єкти або списки чисел.

Завдання циклу For Next Loop полягає в тому, щоб виконувати однакові дії (рядки коду) для кожного елемента в колекції.

Приклади колекцій:

- Клітини в асортименті.
- Робочі листи в робочому зошиті.
- Відкриті робочі зошити на комп'ютері.
- Зведені таблиці на аркуші.
- Зведені поля в зведеній таблиці.
- Форми на листі.

- І будь-який інший об'єкт, з яким ви взаємодієте в Excel.»

Крок 5: Отримання теоретичного матеріалу у вигляді зображення:

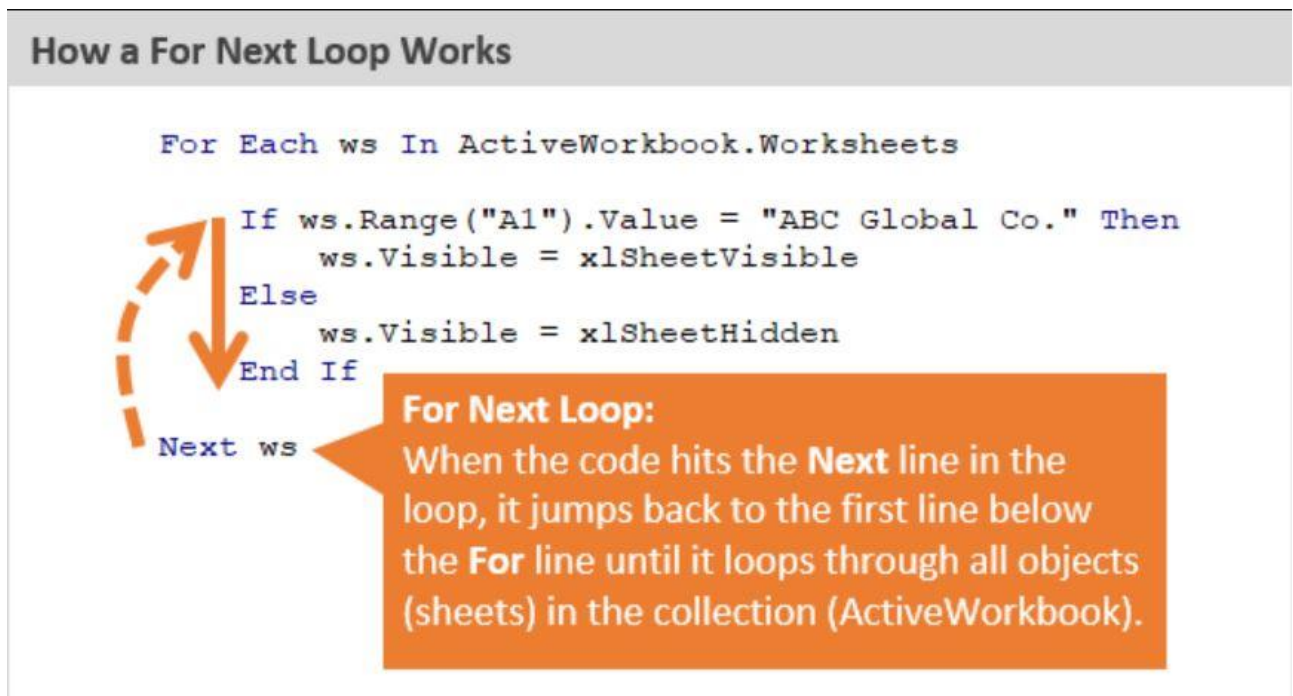


Рисунок 3.1 - Теоретичний матеріал у вигляді зображення

Крок 6: Отримання теоретичного матеріалу:

«У наведеному нижче прикладі міститься цикл For Next, який проходить по кожному аркушу в книзі і відображає кожен лист. Цикл починається з першого елемента в колекції (перший лист в робочій книзі) і виконує рядок коду між рядками For і Next для кожного елемента в колекції (кожен лист в робочій книзі).

```

Sub Unhide_Multiple_Sheets()
    Dim ws As Worksheet
    For Each ws In
        ActiveWorkbook.Worksheets
    ws.Visible = xlSheetVisible
    Next ws
End Sub»
  
```

Крок 7: Отримання теоретичного матеріалу:

«Звичайно, ми можемо використовувати логічні оператори, такі як оператори If, для перевірки властивостей і умов перед виконанням дій.

Наступний макрос відображає тільки ті листи, які мають фразу «ABC Global Co.» в осередку A1 кожного листа, і приховує всі інші листи.

```
Sub Unhide_Report_Sheets()
Dim ws As Worksheet
For Each ws In ActiveWorkbook.Worksheets
If ws.Range("A1").Value = "ABC Global Co." Then
ws.Visible = xlSheetVisible
Else
ws.Visible = xlSheetHidden
End If
Next ws
End Sub»
```

Крок 8: Отримання теоретичного матеріалу:

«Роглянемо The For Each Next Loop: цикл по колекції елементів.

Як ми бачили раніше, цикл «The For Each Next Loop» дозволяє нам перебирати колекцію предметів або об'єктів. Це, напевно, найпоширеніший цикл, який ми використовуємо в Excel, тому що ми працюємо з колекціями об'єктів. Знову ж таки, ці колекції є комірки в діапазоні, робочі таблиці в робочій книзі, зведені таблиці в робочій таблиці і т.д.»

Крок 9: Отримання теоретичного матеріалу у вигляді зображення:

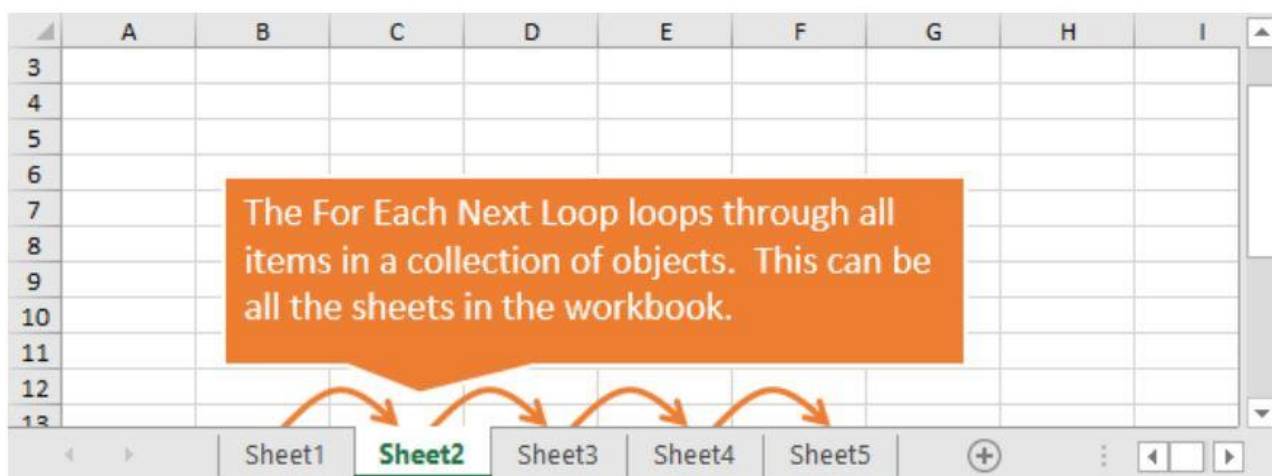


Рисунок 3.2 - Теоретичний матеріал у вигляді зображення

«Ми будемо використовувати приклад написання циклу «For Each Next Loop», щоб переглянути всі робочі листи в робочій книзі.»

Крок 10: Отримання теоретичного матеріалу:

«Існує чотири основні кроки для написання циклу For Each Next в VBA:

- Оголосіть змінну для об'єкта.
- Напишіть для кожного рядка змінну і колекцію посилань.
- Додайте рядок (і) коду для повтору для кожного елемента в колекції.
- Напишіть наступний рядок, щоб закрити цикл.»

Крок 11: Отримання теоретичної інформації:

«Цикл For ... Next в VBA Excel призначений для виконання групи операторів необхідну кількість разів, задане керуючої змінної циклу - лічильником. При виконанні циклу значення лічильника після кожної ітерації збільшується або зменшується на число, вказане виразом оператора Step, або, за замовчуванням, на одиницю. Коли необхідно застосувати цикл до елементів, кількість яких і індексація в групі (діапазон, масив, колекція) невідомі, слід використовувати цикл For Each ... Next.»

Крок 12: Після закінчення роботи з теоретичними матеріалами користувач переходить до практичних завдань.

Крок 13: Отримання практичного завдання:

«Для чого використовуються макроси в Excel VBA?

Варіанти відповіді:

1. Для виконання складних дій;
2. Для виконання однотипних дій;
3. Для роботи з вкладеними циклами.

Правильна відповідь - № 2»

Крок 14: Отримання практичного завдання:

«Завдання циклу For Next Loop полягає в тому, щоб ...?

Варіанти відповіді:

1. Виконувати однакові дії (рядки коду) для кожного елемента в колекції;
2. Перебирати колекцію предметів або об'єктів;

3. Переглядати колекцію, елементами якої можуть бути об'єкти або списки чисел.

Правильна відповідь - № 1»

Крок 15: Отримання практичного завдання:

«Завдання циклу The For Each Next Loop полягає в тому, щоб ...?»

Варіанти відповіді:

1. Виконувати однакові дії (рядки коду) для кожного елемента в колекції;
2. Перебирати колекцію предметів або об'єктів;
3. Переглядати колекцію, елементами якої можуть бути об'єкти або списки чисел.

Правильна відповідь - № 2»

Крок 16: Отримання практичного завдання:

«Завдання циклу For Next полягає в тому, щоб ...?»

Варіанти відповіді:

1. Виконувати однакові дії (рядки коду) для кожного елемента в колекції;
2. Перебирати колекцію предметів або об'єктів;
3. Переглядати колекцію, елементами якої можуть бути об'єкти або списки чисел.

Правильна відповідь - № 3»

Крок 17: Отримання практичного завдання:

«Скільки існує основних кроків для написання циклу For Each Next в VBA?»

Варіанти відповіді:

1. 3;
2. 4;
3. 5.

Правильна відповідь - № 2»

Крок 18: Отримання практичного завдання:

«Що є першим елементом в колекції при виконанні циклу For Next?»

Варіанти відповіді:

1. Перший лист в робочій книзі;
2. Перший елемент в документі;
3. Будь-який обраний елемент.

Правильна відповідь - № 1»

Крок 19: Отримання практичного завдання:

«Для чого використовуються цикли у VBA?

Варіанти відповіді:

1. Для виконання певного блоку програми
2. Для виконання певної дії певну кількість разів
3. Для виконання змін у налаштуваннях Екселя.

Правильна відповідь - № 2»

Крок 20: Отримання практичного завдання:

«Для чого використовується цикл For_Next?

Варіанти відповіді:

1. Перебирання послідовності значень для змінної-лічильника
2. Перебирання послідовності значень для константи
3. Для виконання певної математичної дії

Правильна відповідь - № 1»

Крок 21: Отримання практичного завдання:

«При роботі з For_Next для змінної виконується певний код що міститься в?

Варіанти відповіді:

1. Тілі циклу For_Next
2. Певній функції, що задана після циклу
3. У відповідній змінній

Правильна відповідь - № 1»

Крок 22: Отримання практичного завдання:

«Який оператор використовується для визначення кроку в циклі?

Варіанти відповіді:

1. Step
2. Krok
3. Next

Правильна відповідь - № 1»

Крок 23: Отримання довідкової інформації до практичного завдання:

«Цикл For..Next

Синтаксис

For counter = Start To End [Step StepSize]

Statements

Next [counter]

Counter - будь-яка чисельна змінна VBA

Start - будь чисельне вираження, визначає початкове значення для змінної counter

End - чисельне вираження, визначає кінцеве значення для змінної counter

Statements - один, кілька або жодного оператора VBA (тіло циклу).

За замовчуванням VBA збільшує змінну counter на 1 кожен раз при виконанні операторів в циклі. Можна задати інше значення (StepSize - будь чисельне вираження), на яке буде змінюватися counter.

Ключове слово Next повідомляє VBA про те, що досягнутий кінець циклу. Необов'язкова змінна counter після ключового слова Next повинна бути тієї ж самої змінної counter, яка була задана після ключового слова For на початку структури циклу.»

Крок 24: Отримання практичного завдання:

Оберіть цикл For..Next для підрахунку чисел від десяти до одного.

Варіанти відповіді:

1. For i = 10 To 1 Step -1

 iArray(i) = i

Next i

2. For i = 10 To 1 Stp -1

 iArray(i) = i

Next i

3. For i = 10 To 1 Step 1

 iArray(i) = i

Next i

Правильна відповідь - № 1»

Крок 25: Отримання довідкової інформації до практичного завдання:

Цикл For..Next

Синтаксис

For counter = Start To End [Step StepSize]

 Statements

Next [counter]

Counter - будь-яка чисельна змінна VBA

Start - будь чисельне вираження, визначає початкове значення для змінної counter

End - чисельне вираження, визначає кінцеве значення для змінної counter

Statements - один, кілька або жодного оператора VBA (тіло циклу).

За замовчуванням VBA збільшує змінну counter на 1 кожен раз при виконанні операторів в циклі. Можна задати інше значення (SterSize - будь чисельне вираження), на яке буде змінюватися counter.

Ключове слово Next повідомляє VBA про те, що досягнутий кінець циклу. Необов'язкова змінна counter після ключового слова Next повинна бути тієї ж самої змінної counter, яка була задана після ключового слова For на початку структури циклу.

Крок 26: Отримання практичного завдання:

«Оберіть цикл For..Next для підрахунку чисел від одного до десяти.

Варіанти відповіді:

1. For i = 1 To 10

SUM = SUM +1

Next i

2. For int = 1 To 10

SUM +1

Next i

3. For int = 10 To 1

SUM +1

Next i

Правильна відповідь - № 1»

Крок 27: Отримання практичного завдання з полем для вводу:

«Запишіть цикл для виводу кожного листа в книзі Екселя. Використайте цикл For Next.

Результатом є наступний код:

«Sub Unhide_Multiple_Sheets()

Dim ws As Worksheet

```

For Each ws In ActiveWorkbook.Worksheets
ws.Visible = xlSheetVisible
Next ws
End Sub»

```

Крок 28: Отримання практичного завдання з полем для вводу:

«Заповнити десять перших комірок першого стовпчика активного листа Excel цифрами від 1 до 10 використовуючи цикл For Next.

Результатом є наступний код:

```

Sub test1()
Dim i As Long
For i = 1 To 10
Cells(i, 1) = i
Next
End Sub»

```

Крок 29: Після закінчення роботи з тренажером користувач отримує доступ до кнопки «Повторити роботу з тренажером?», що дозволяє перейти до першого навчального елементу в тренажері.

3.3 Блок-схема програми-тренажера

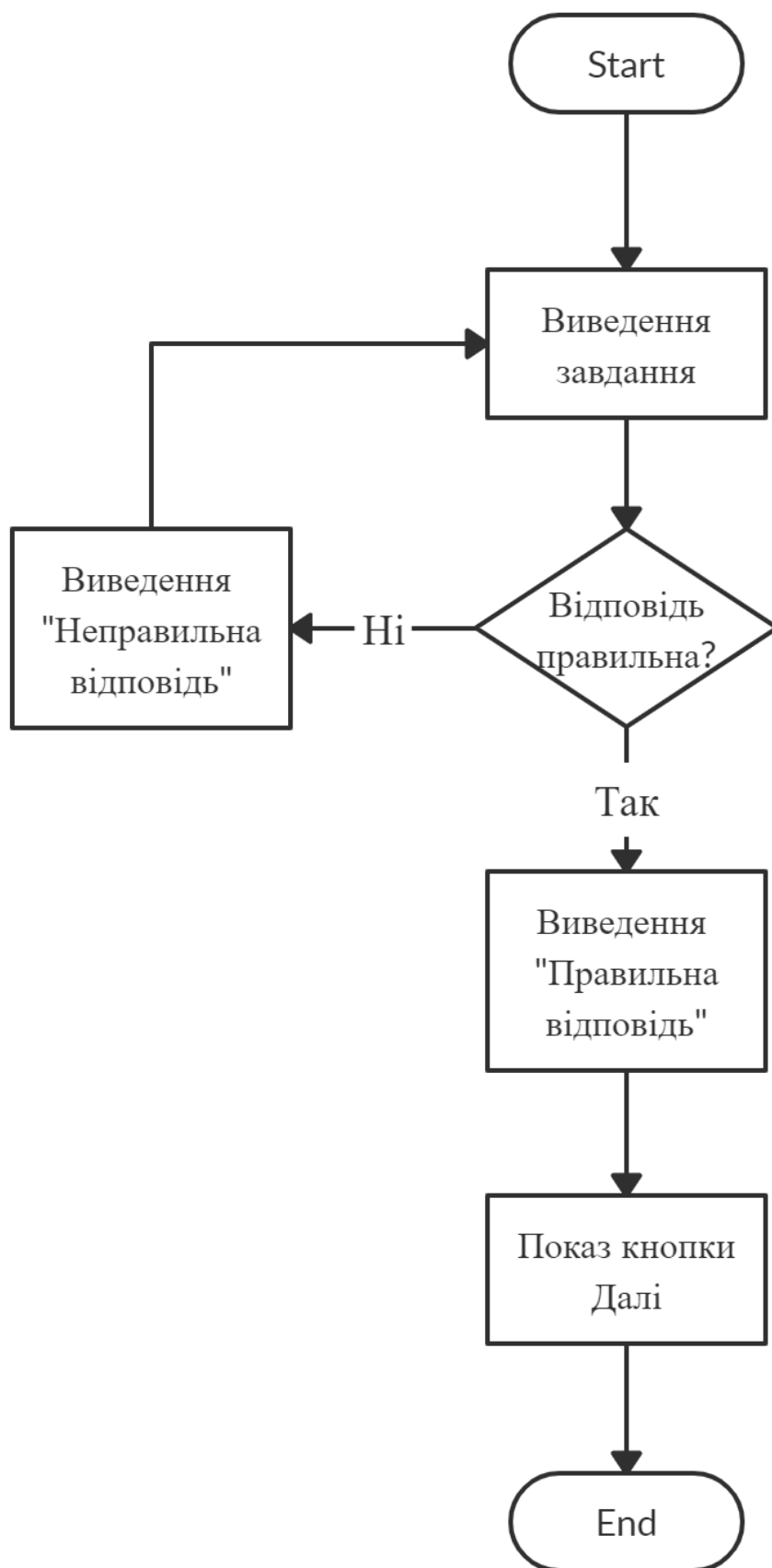


Рисунок 3.3 – Блок-схема програми-тренажера

4 Практична частина

4.1 Опис програмного забезпечення

Після запуску програмного забезпечення користувач переходить до екрану з вибором первинних налаштувань програми де може обрати розширення екрану та якість зображення для будь яких комп'ютерів.

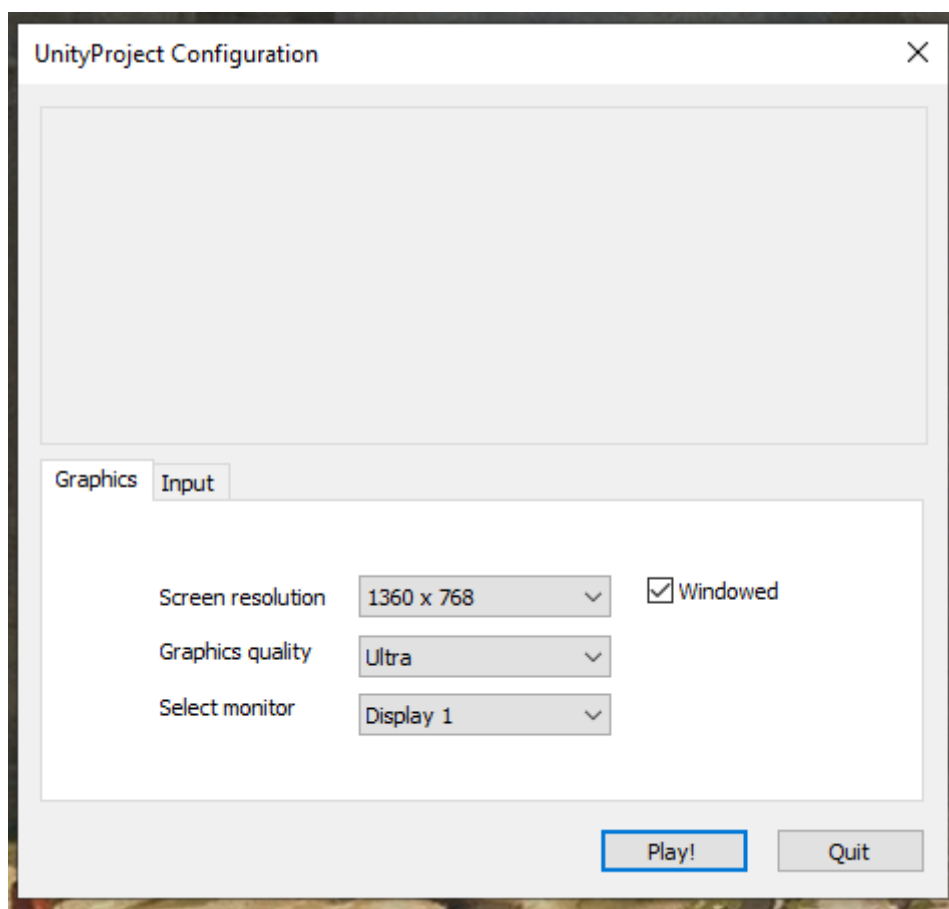


Рисунок 4.1 – Вибір первинних налаштувань тренажеру

Після вибору налаштувань та натискання на кнопку «Play!» користувач переходить до роботи з тренажером.

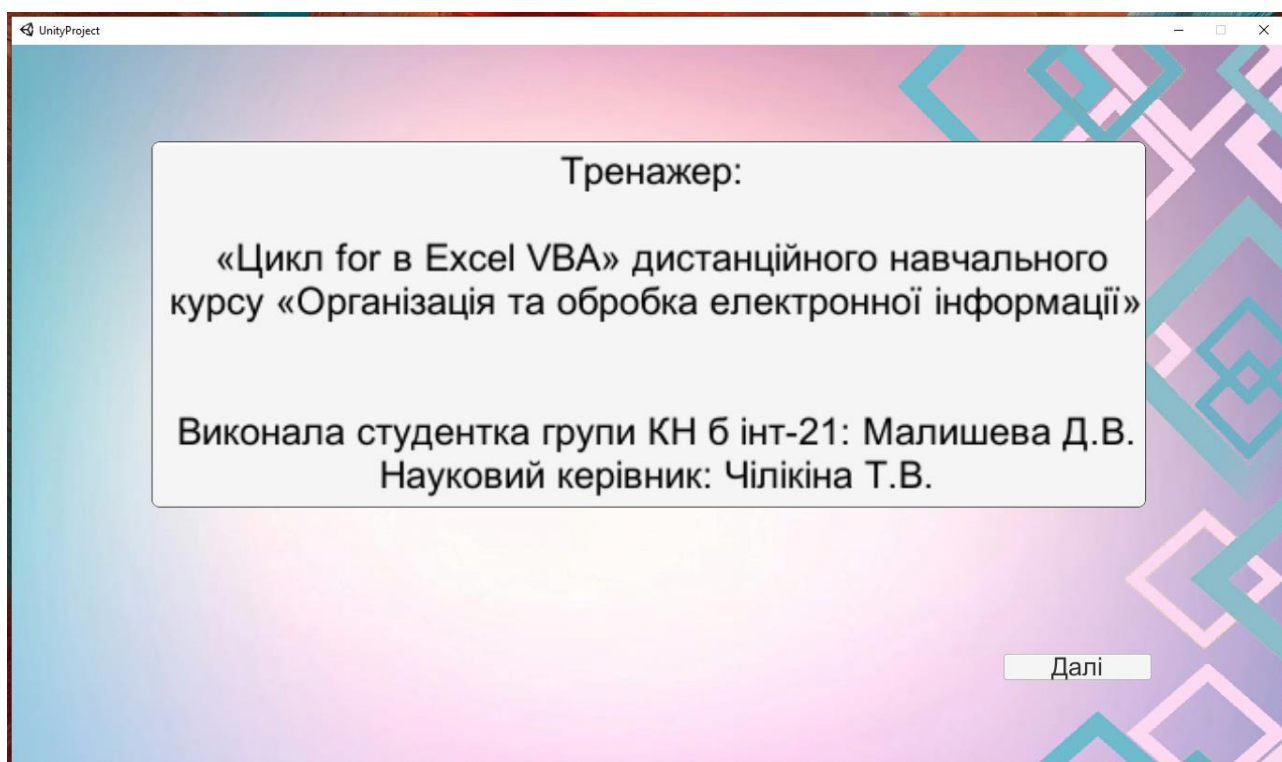


Рисунок 4.2 – Початок роботи з тренажером

Після натискання на кнопку «Далі» користувач переходить до практичних завдань тренажеру в яких розміщено коротку теорію та тести.

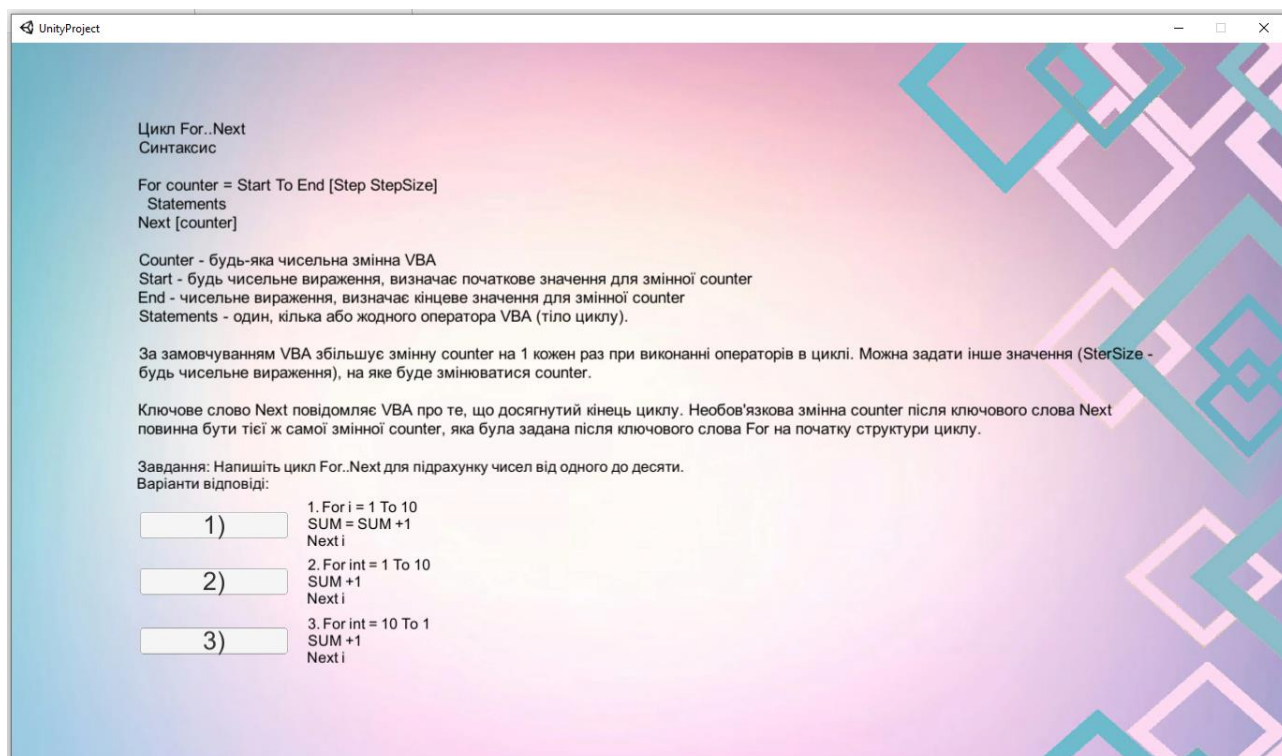


Рисунок 4.3 – Зовнішній вигляд практичних завдань тренажера

Після вибору неправильної відповіді виводиться повідомлення «Неправильна відповідь». Закрити це повідомлення можна натиснувши на нього.

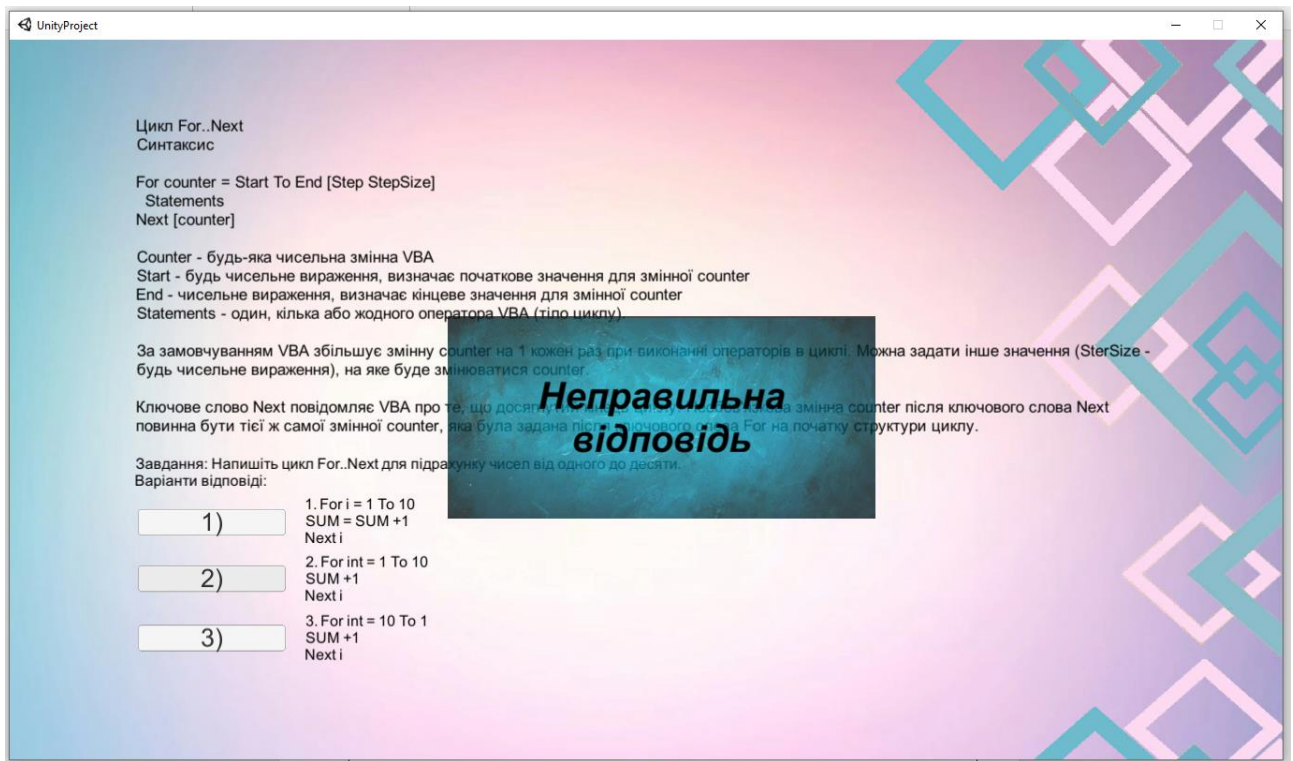


Рисунок 4.4 – Виведення повідомлення про помилку

Після вибору правильної відповіді виводиться повідомлення «Правильна відповідь» та з'являється кнопка «Далі». Закрити це повідомлення можна натиснувши на нього.

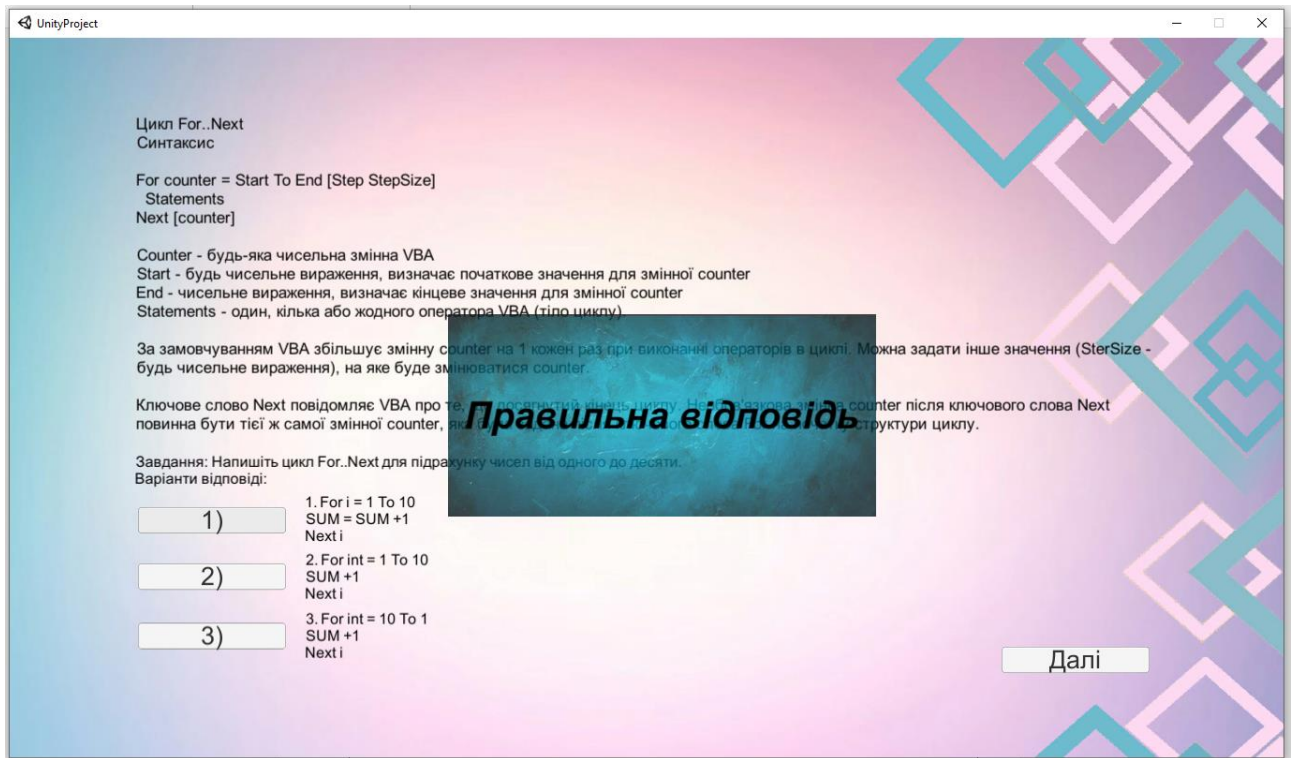


Рисунок 4.5 – Виведення повідомлення про правильну відповідь

Після закінчення роботи з практичними завданнями кнопка «Далі» слугує для повтору роботи з тренажером, перенаправляючи користувача на перший екран.

4.2 Перевірка валідності

Вікно програми після запуску тренажера:

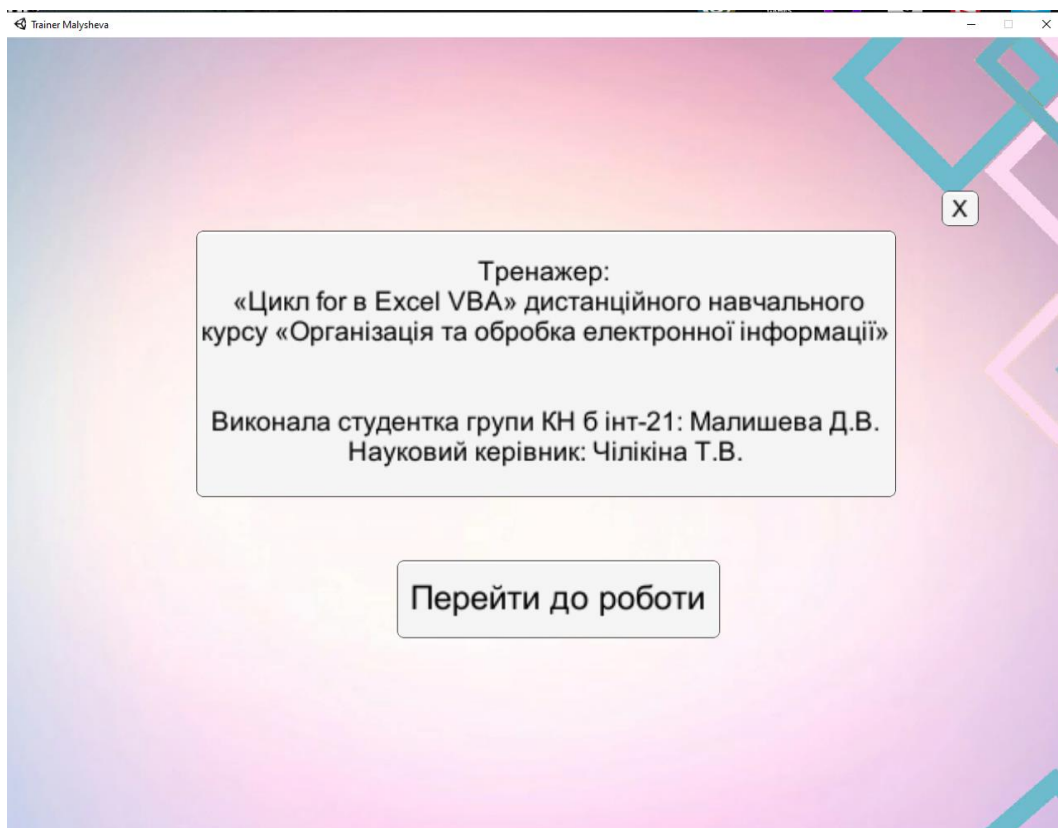


Рисунок 4.6 – Головне меню тренажера

Вікно програми після переходу до теоретичних матеріалів:

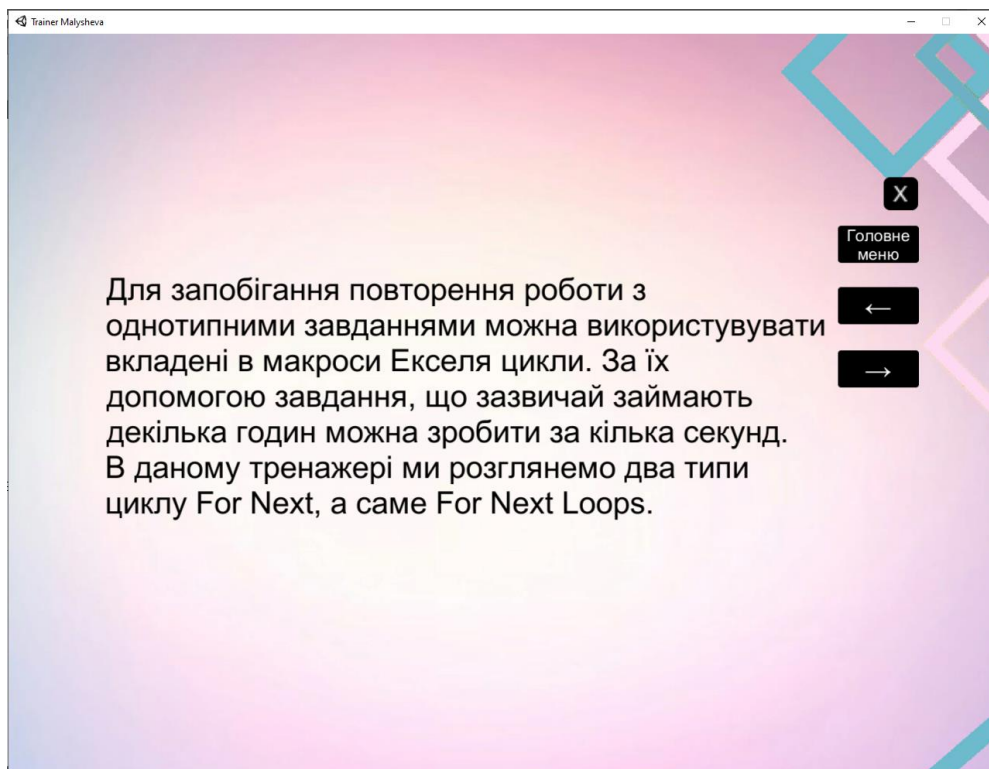


Рисунок 4.7 – Теоретичні матеріали тренажера

Теоретичні матеріали у вигляді зображення:



Рисунок 4.8 – Теоретичні матеріали у вигляді зображення

Робота з завданнями тестового типу:

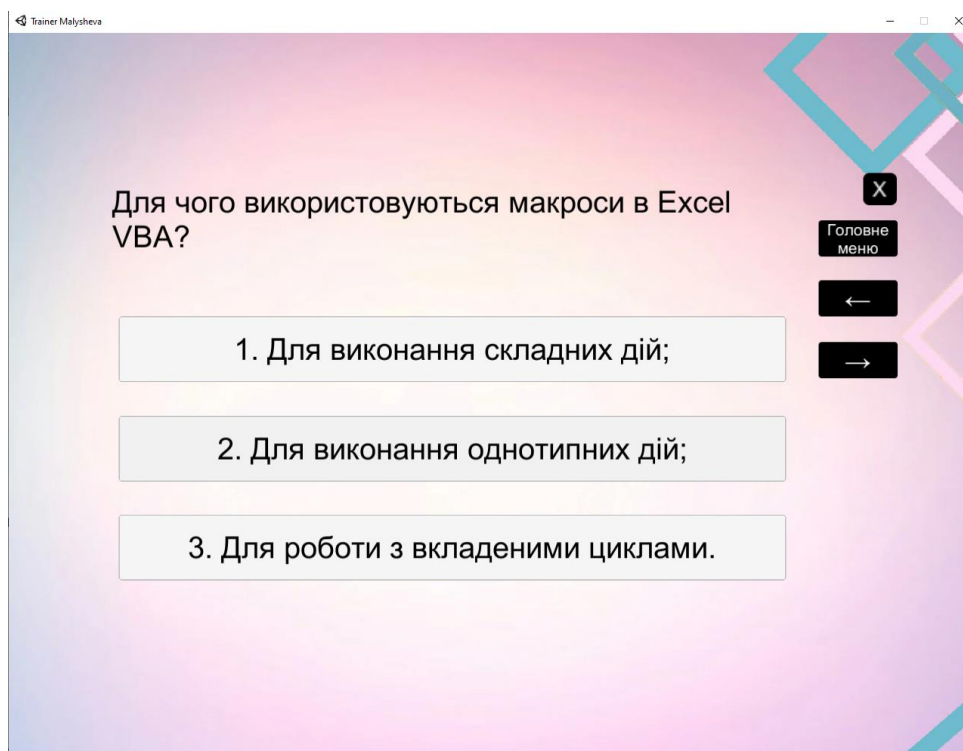


Рисунок 4.8 – Тестове завдання

Вигляд завдання тестового типу після вибору неправильної відповіді:

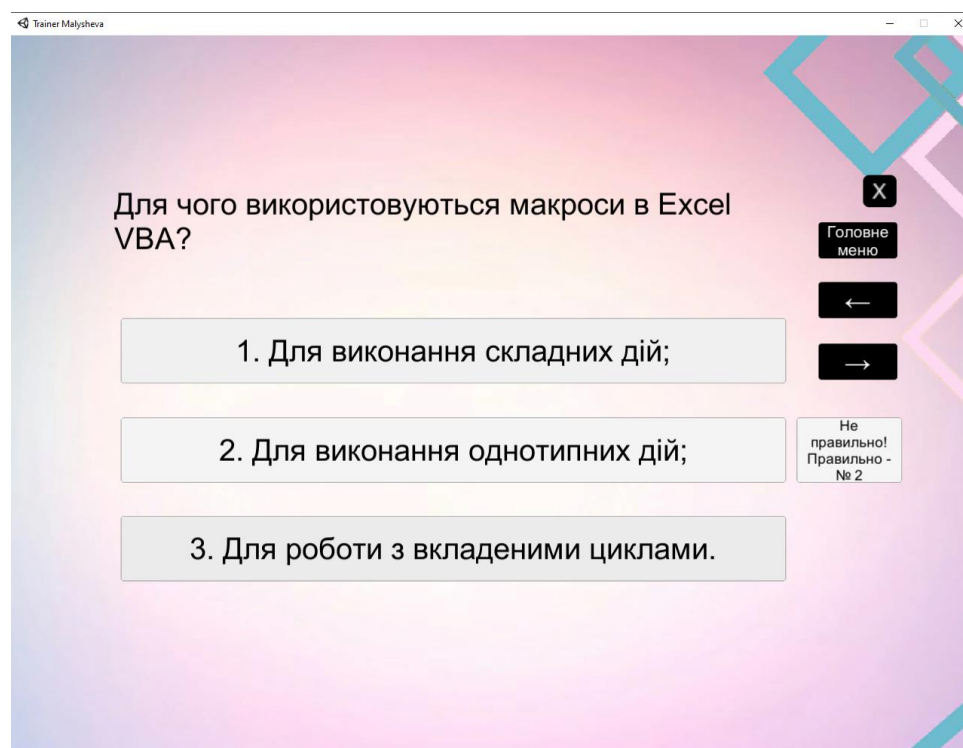


Рисунок 4.9 – Тестове завдання після вибору неправильної відповіді

Вигляд завдання тестового типу після вибору правильної відповіді:

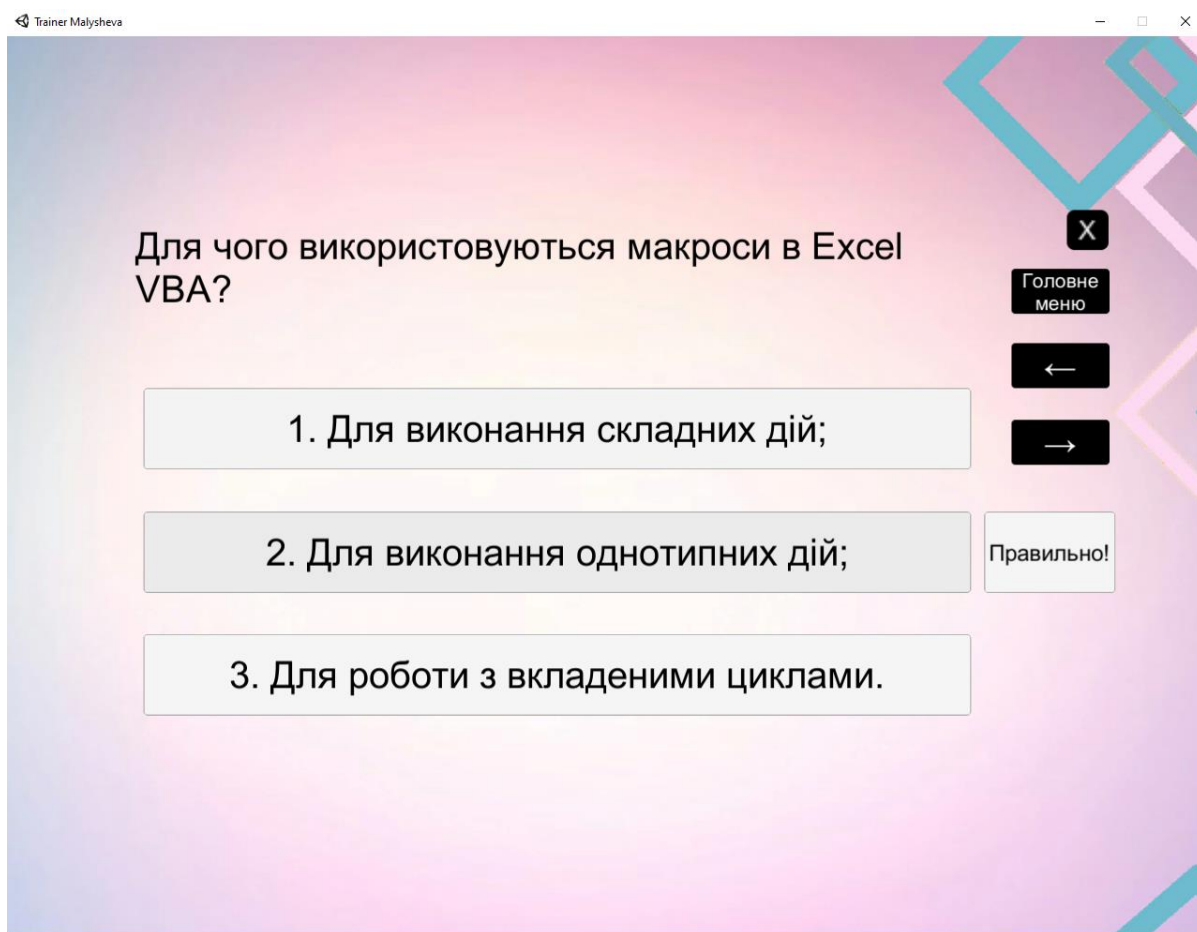


Рисунок 4.9 – Тестове завдання після вибору правильної відповіді

Зовнішній вигляд практичного завдання:

Цикл For..Next
Синтаксис

```
For counter = Start To End [Step StepSize]
  Statements
Next [counter]
```

Counter - будь-яка чисельна змінна VBA
 Start - будь чисельне вираження, визначає початкове значення для змінної counter
 End - чисельне вираження, визначає кінцеве значення для змінної counter
 Statements - один, кілька або жодного оператора VBA (тіло циклу).

За замовчуванням VBA збільшує змінну counter на 1 кожен раз при виконанні операторів в циклі. Можна задати інше значення (StepSize - будь чисельне вираження), на яке буде змінюватися counter.

Ключове слово Next повідомляє VBA про те, що досягнутий кінець циклу. Необов'язкова змінна counter після ключового слова Next повинна бути тієї ж самої змінної counter, яка була задана після ключового слова For на початку структури циклу.

Оберіть цикл For..Next для підрахунку чисел від десяти до одного.

1. For i = 10
To 1 Step -1
iArray(i) = i
Next i

2. For i = 10
To 1 Stp -1
iArray(i) = i
Next i

3. For i = 10
To 1 Step 1
iArray(i) = i
Next i

Рисунок 4.10 – Практичне завдання

Зовнішній вигляд практичного завдання після вибору неправильної відповіді:

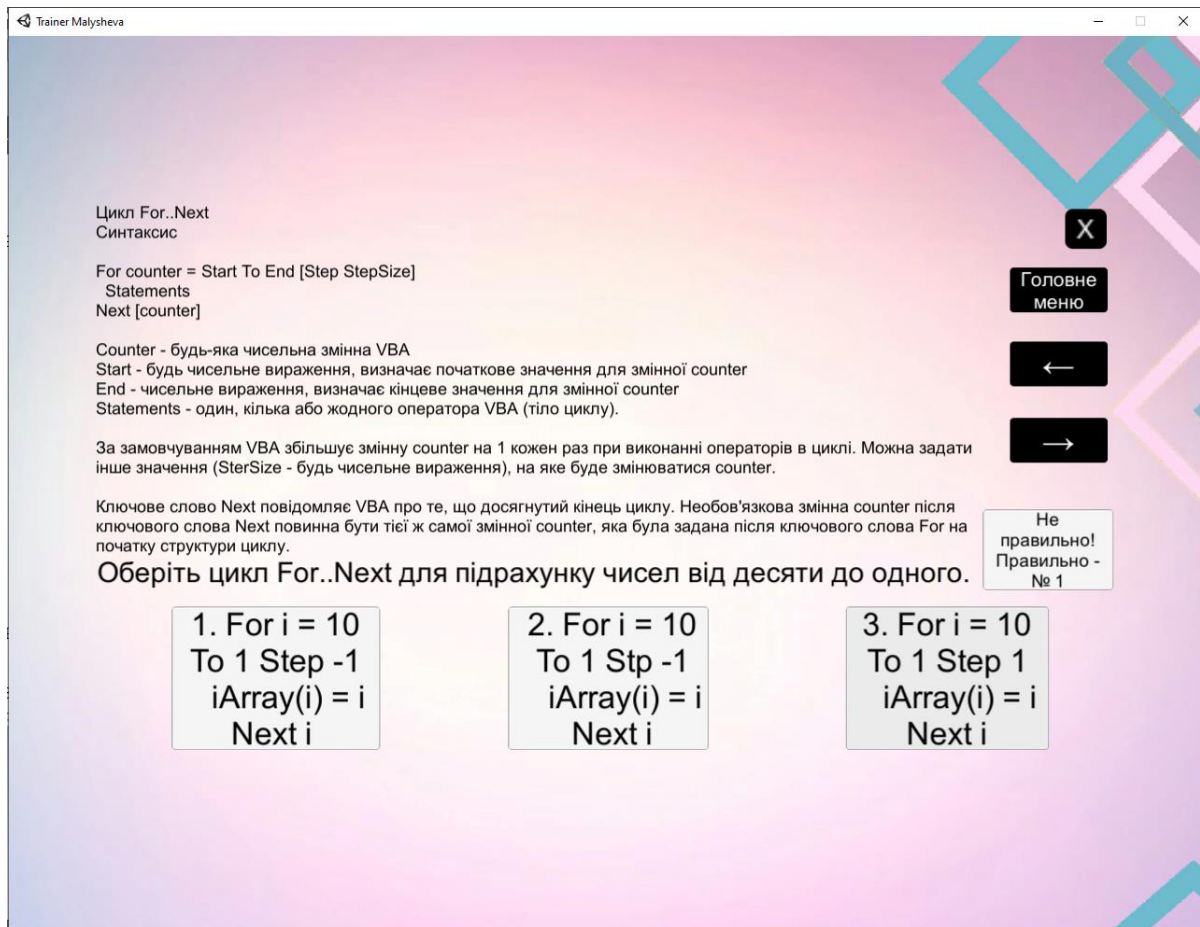


Рисунок 4.11 – Практичне завдання після вибору неправильної відповіді

Зовнішній вигляд практичного завдання після вибору правильної відповіді:

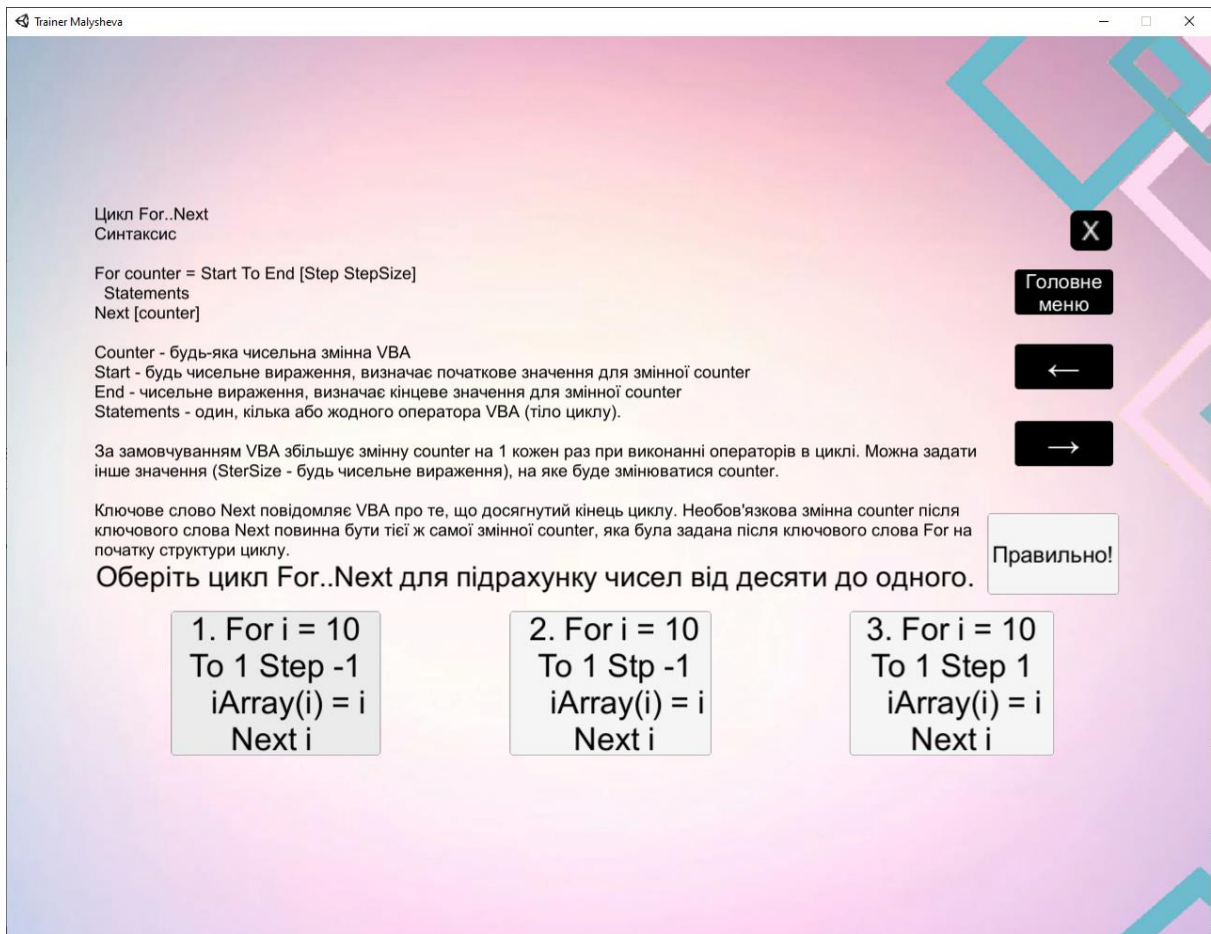


Рисунок 4.12 – Практичне завдання після вибору правильної відповіді

Зовнішній вигляд практичного завдання з полем для вводу:

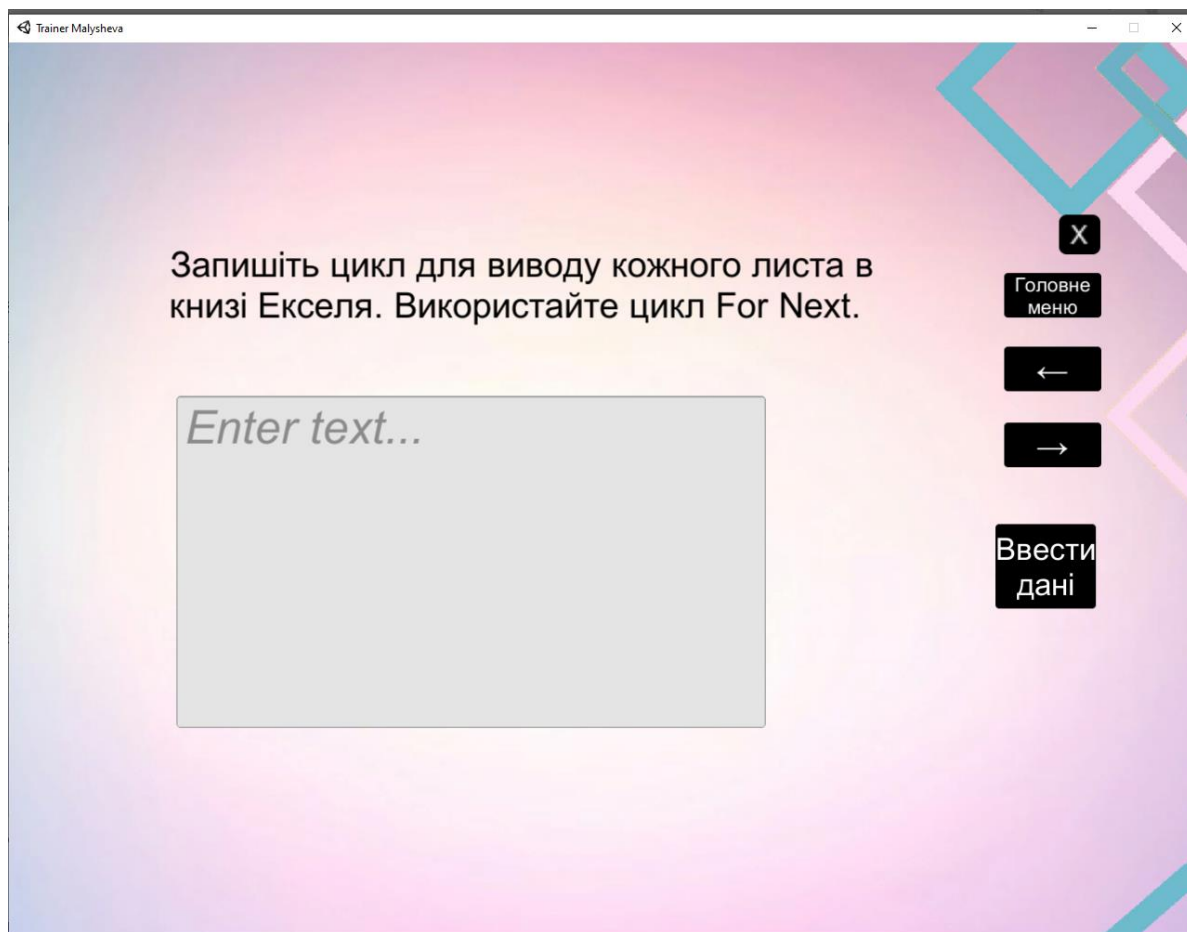


Рисунок 4.13 – Практичне завдання з полем для вводу

Зовнішній вигляд практичного завдання з полем для вводу після вибору відповіді:

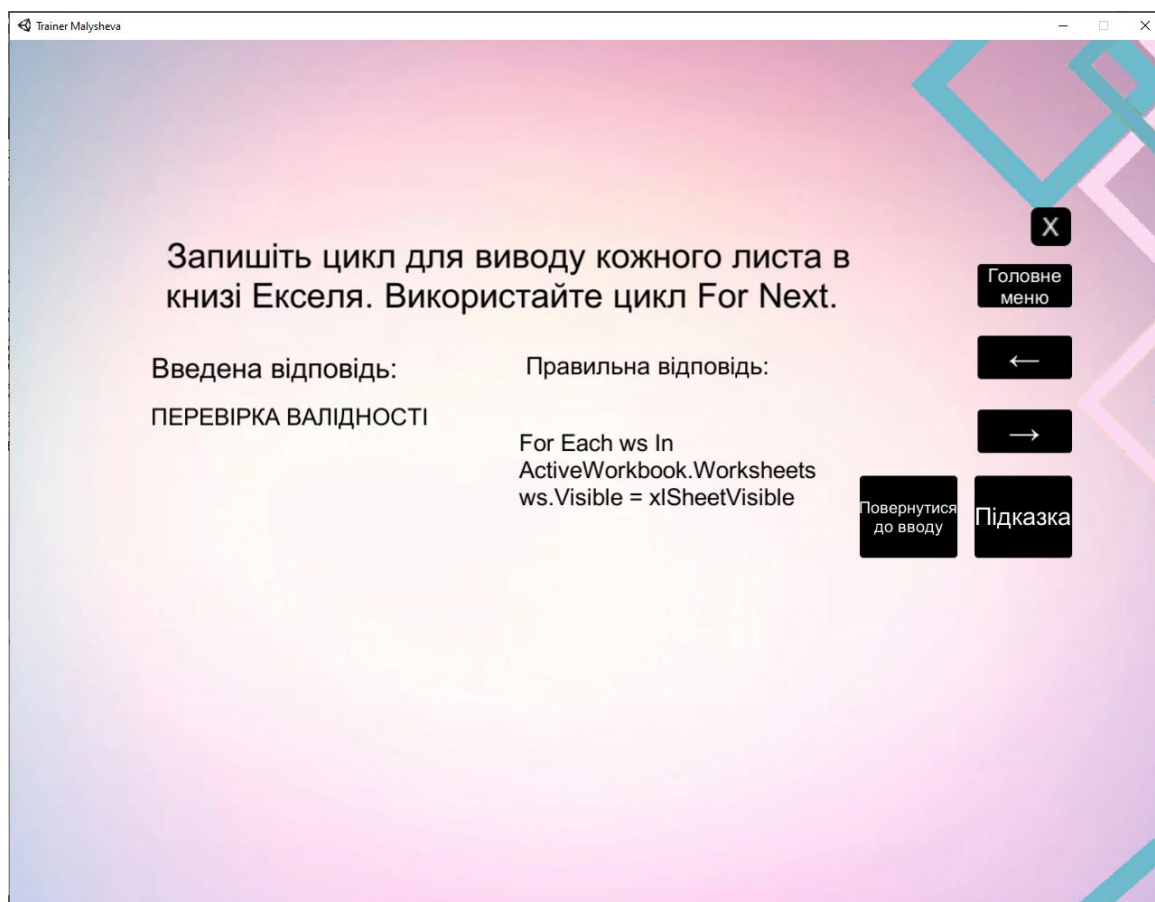


Рисунок 4.14 – Практичне завдання з полем для вводу після вибору відповіді

Зовнішній вигляд практичного завдання з полем для вводу після вибору підказки:

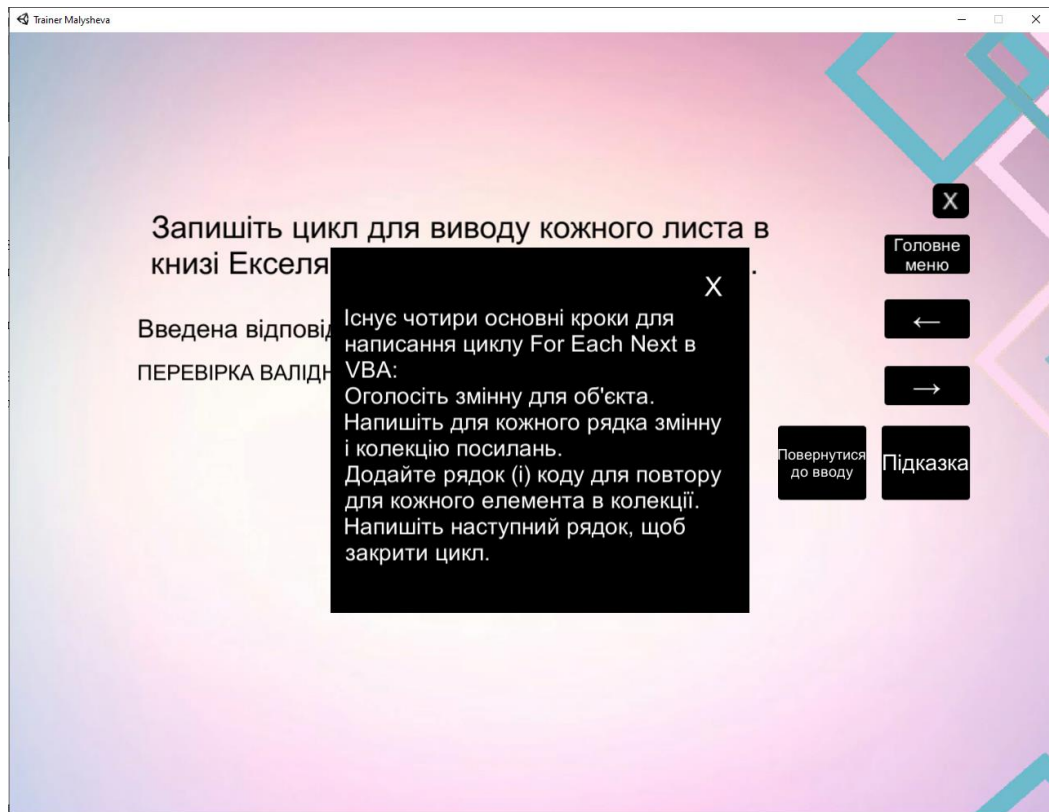


Рисунок 4.15 – Практичне завдання з полем для вводу після вибору підказки

Вікно тренажера після завершення роботи з його навчальними матеріалами:

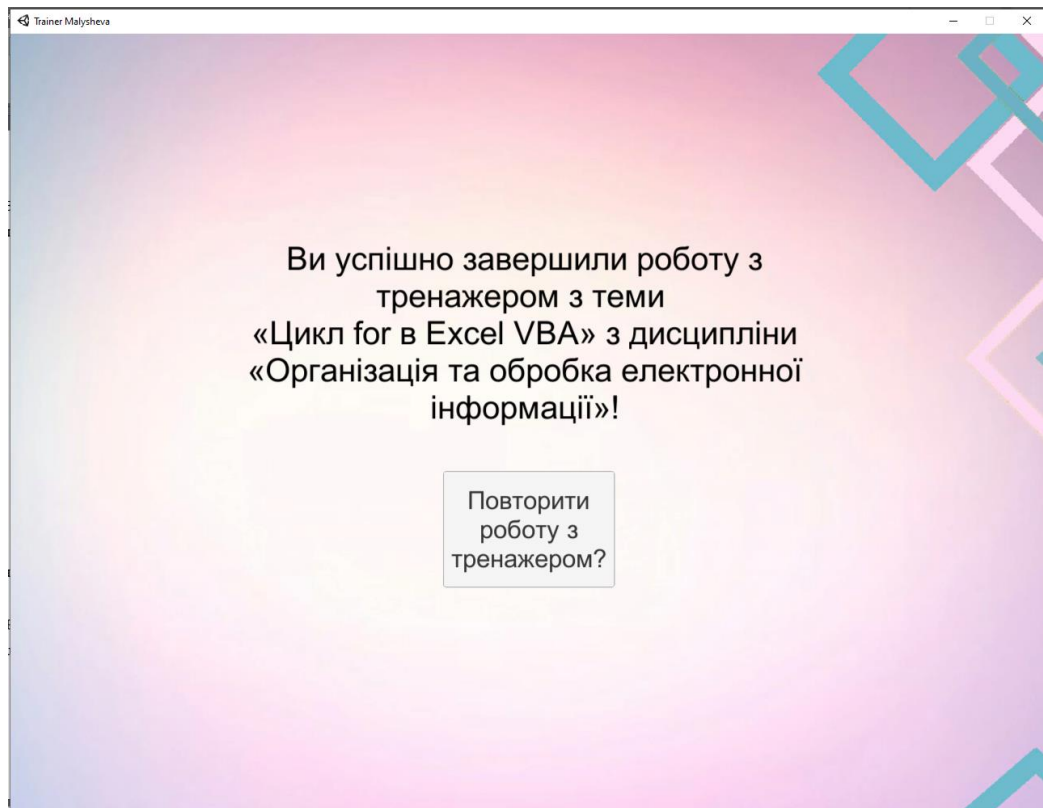


Рисунок 4.16 – Завершення роботи з тренажером

Також існує можливість завершити роботу з тренажером в будь-який момент роботи з ним, через натиснення на кнопку «Головне меню». Таким чином можна повторити роботу з тренажером в цілому.



Рисунок 4.17 – Кнопки навігації та кнопка «Головне меню»

ВИСНОВКИ

Результатами бакалаврської роботи є:

- Створено елементи навчального тренажеру з теми «Цикл for в Excel VBA»;
- Створено алгоритм роботи тренажеру;
- Створено блок-схему роботи алгоритму;
- Протестовано роботу тренажеру.

Плюсами розробленого програмного забезпечення є:

- Наявність короткої теорії та практичних завдань на одному екрані;
- Перевірка введеної відповіді;
- Можливість закриття повідомлення про правильність відповіді натисканням на неї;
- Приємний дизайн.

Мету та завдання бакалаврської роботи виконано, розроблено елементи програмного забезпечення у вигляді тренажеру з теми «Цикл for в Excel VBA». Тренажер розроблено на платформі Unity Engine. з використанням мови програмування C# та Microsoft Visual Studio.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Тітар Є.В. Пояснювальна записка до бакалаврської роботи на тему «Розробка тренажеру з теми «Визначення рангу елементу системи» дисципліни «Теорія систем та математичне моделювання»/ Тітар Є.В. [Електронний ресурс].— Режим доступу до ресурсу: <http://dspace.puet.edu.ua/handle/123456789/9027>
2. Цикозін О.В. Пояснювальна записка до бакалаврської роботи на тему «Розробка тренажеру на тему «Програмна реалізація та алгоритмізація тренажера дистанційного навчального курсу «Обчислювальні методи» з теми «розв’язування диференціальних методом Ейлера»/ Цикозін О.В. [Електронний ресурс].— Режим доступу до ресурсу: <http://dspace.puet.edu.ua/handle/123456789/9029>
3. Оператор цикла «For» в Visual Basic [Електронний ресурс].— Режим доступу до ресурсу: <https://office-guru.ru/excel/cikly-v-vba-463.html#ForLoop>
4. VBA Excel. Цикл For... Next [Електронний ресурс].— Режим доступу до ресурсу: <https://vremya-ne-zhdet.ru/vba-excel/tsikl-for-next/>
5. Циклы VBA (ч.3). Вложенные циклы [Електронний ресурс].— Режим доступу до ресурсу: <http://on-line-teaching.com/vba/lsn0118.html>
6. Циклы VBA (ч.1). Команды организации циклов [Електронний ресурс].— Режим доступу до ресурсу: <http://on-line-teaching.com/vba/lsn0116.html#%D0%9D%D0%B0%D1%87%D0%B0%D0%BB%D0%BE%20%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D1%8B>
7. Unity (Рушій гри) [Електронний ресурс].— Режим доступу до ресурсу: [https://uk.wikipedia.org/wiki/Unity_\(%D1%80%D1%83%D1%88%D1%96%D0%B9_%D0%B3%D1%80%D0%B8\)](https://uk.wikipedia.org/wiki/Unity_(%D1%80%D1%83%D1%88%D1%96%D0%B9_%D0%B3%D1%80%D0%B8))
8. IDE Visual Studio [Електронний ресурс].— Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio

9. IDE Visual Studio [Электронный ресурс].– Режим доступа до ресурсу: <http://www.codenet.ru/progr/vbasic/Visual-Basic-NET/>
10. Working with scripts [Электронный ресурс].– Режим доступа до ресурсу: <https://learn.unity.com/tutorial/working-with-scripts?language=en#>
11. Р. Ш. Гайнанова, О. А. Широкова, Программирование на Visual Basic for Applications в Excel: учебное пособие /Р. Ш. Гайнанова, О. А. Широкова – Казань: КФУ, 2012. – 153с
12. Культин Н.Б., Цой Л.Б., Visual Basic для студентов и школьников /Культин Н.Б., Цой Л.Б., БХВ-Петербург, 2010, 416 с.

ДОДАТОК А

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine.UI;
using UnityEngine;

public class Inputs : MonoBehaviour
{
    public GameObject Input;
    public GameObject AfterInput;

    public InputField inputedTxt;
    public Text ShowInpTxt;

    public void Next()
    {
        Input.SetActive(false);
        AfterInput.SetActive(true);

        ShowInpTxt.text = inputedTxt.text;
    }
}

public class QuitButton : MonoBehaviour
{
    public void QuitGame()
    {

```

```

    Debug.Log ("QUIT!!");
    Application.Quit();
}

```

```

void Start()
{
    Button btn = nxtButton.GetComponent<Button>();
    btn.onClick.AddListener(TaskOnClick);
}

```

```

void TaskOnClick()
{
    Theme1.SetActive(false);
    Theme2.SetActive(true);

    Debug.Log("You touched this button.");
}

```

```

void Start()
{
    Button btn = menuButton.GetComponent<Button>();
    btn.onClick.AddListener(TaskOnClick);
}

```

```

void TaskOnClick()
{
    Theme6.SetActive(false);
    MainMenu.SetActive(true);
}

```



```
    Debug.Log("You touched this button.");  
}
```

```
public void Next()  
{  
    Input.SetActive(false);  
    AfterInput.SetActive(true);  
  
    ShowInpTxt.text = inputedTxt.text;  
}  
}
```